

Méthodes de décomposition pour la programmation linéaire en nombres entiers

Christian Artigues

LAAS-CNRS

6 mai 2014

Optimisation discrète ou combinatoire

Trouver le minimum (ou le maximum) d'une fonction f définie sur un ensemble discret \mathcal{X} .

$$\min f(x), x \in \mathcal{X}$$

Problème du voyageur de commerce

Un livreur (ou un voyageur de commerce) doit desservir n villes en partant de la ville 1 et revenir à son point de départ. c_{ij} est le coût de déplacement de la ville i et la ville j . Quelle est la tournée de coût minimal ?

Problème du sac à dos

Une entreprise dispose d'un budget total b pour financer certains projets parmi un ensemble de n projets. Chaque projet i a un coût a_i et un profit attendu c_i . Quels projets l'entreprise doit-elle financer pour maximiser son profit total ?

Problème de couverture d'ensemble

Les communications d'un ensemble de m régions doivent être couvertes par des antennes à installer sur n sites prédéterminés. Chaque site i a un coût d'installation c_i et peut couvrir un ensemble de régions S_i . Quelles antennes installer de façon à couvrir l'ensemble des régions et à minimiser le coût total d'installation ?

Problème de localisation d'entrepôts

Une entreprise livre ses marchandises à m clients et dispose pour cela d'un ensemble de n dépôts potentiels. L'utilisation d'un dépôt j entraîne un coût fixe f_j et le transport de la commande d'un client i à partir d'un dépôt j entraîne un coût c_{ij} . Quels dépôts ouvrir et quel plan d'acheminement établir pour livrer tous les clients en minimisant la somme des coûts fixes et variables ?

Problème d'ordonnancement à une machine

Une ligne d'assemblage doit assembler n produits. Chaque produit i possède une durée d'assemblage p_i , une date d'arrivée des composants sur la ligne r_i et une date de livraison prévue d_i . Sachant que la ligne ne peut assembler qu'un produit à la fois, quelles sont les dates de début d'assemblage de chaque produit qui minimisent le plus grand retard de livraison ?

Problème d'affectation

n travaux doivent être réalisés par n employés. Certaines personnes sont plus expérimentées que d'autre et le temps de réalisation par un employé i d'un travail j a une durée c_{ij} . Quelle est l'affectation qui minimise la durée moyenne des tâches ?

Problème de plus court chemin

Un réseau routier est représenté par un graphe orienté $G = (V, A)$. Le transport direct d'un lieu i vers un lieu j est possible si $(i, j) \in A$ et a une durée c_{ij} . Etant donné deux lieux s et t , quel est le chemin à coût minimum de s à t ?

Problème de flot à coût minimum

Un réseau de transport de marchandises est représenté par un graphe orienté $G = (V, A)$. Chaque sommet $i \in V$ est associé à une valeur b_i appelée demande et correspond à un lieu de production ($b_i > 0$), de consommation ($b_i < 0$) ou de simple transit ($b_i = 0$) de marchandises. Chaque arc $(i, j) \in A$ représente un moyen de transport de marchandises entre deux lieux limité par une capacité h_{ij} et de coût unitaire c_{ij} . Quel est le plan de transport qui satisfait toutes les demandes à coût minimal ?

Programmation linéaire en nombres entiers (PLNE)

- Chaque variable de décision est définie sur l'ensemble des réels (\mathbb{R}) ou des entiers ($\mathbb{N}=\{1,2,\dots,n\}$);
- La fonction objectif est linéaire
 $f(x_1, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$ ($f(x) = cx$);
- Les m contraintes sont linéaires $C_j : a_{1j}x_1 + a_{2j}x_2 + \dots + a_{nj}x_n \leq b_j$
($Ax \leq B$)

Formulation 0-1 pour le problème du voyageur de commerce

Un livreur (ou un voyageur de commerce) doit desservir n villes en partant de la ville 1 et revenir à son point de départ. c_{ij} est le coût de déplacement de la ville i et la ville j . Quelle est la tournée de coût minimal ?

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

$$\sum_{j=1, \dots, n, j \neq i} x_{ij} = 1 \quad i = 1, \dots, n$$

$$\sum_{i=1, \dots, n, i \neq j} x_{ij} = 1 \quad j = 1, \dots, n$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad S \subset \{1, \dots, n\}, 2 \leq |S| \leq n - 1$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, n, j = 1, \dots, n, i \neq j$$

Formulation 0-1 pour le problème du sac-à-dos

Une entreprise dispose d'un budget total b pour financer certains projets parmi un ensemble de n projets. Chaque projet j a un coût a_j et un profit attendu c_j . Quels projets l'entreprise doit-elle financer pour maximiser son profit total ?

$$\max \sum_{j=1}^n c_j x_j$$

$$\sum_{j=1}^n a_j x_j \leq b$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n$$

Formulation 0-1 pour le problème de couverture d'ensemble

Les communications d'un ensemble de m régions doivent être couvertes par des antennes à installer sur n sites prédéterminés. Chaque site j a un coût d'installation c_j et peut couvrir un ensemble de régions S_j . Quelles antennes installer de façon à couvrir l'ensemble des régions et à minimiser le coût total d'installation ?

$$\begin{aligned} \min \quad & \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j \geq 1 \\ & x_j \in \{0, 1\} \quad j = 1, \dots, n \end{aligned}$$

avec $a_{ij} = 1$ si $i \in S_j$ et $a_{ij} = 0$ sinon (*matrice d'incidence*).

Formulation 0-1 pour le problème de localisation

Une entreprise livre ses marchandises à m clients et dispose pour cela d'un ensemble de n dépôts potentiels. L'utilisation d'un dépôt j entraîne un coût fixe f_j et le transport de la commande d'un client i à partir d'un dépôt j entraîne un coût c_{ij} . Quels dépôts ouvrir et quel plan d'acheminement établir pour livrer tous les clients en minimisant la somme des coûts fixes et variables ?

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n f_j y_j$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, m$$

$$x_{ij} \leq y_j \quad i = 1, \dots, m; j = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m, j = 1, \dots, n$$

$$y_j \in \{0, 1\} \quad j = 1, \dots, n$$

Formulation mixte pour le problème d'ordonnancement à une machine

Une ligne d'assemblage doit assembler n produits. Chaque produit i possède une durée d'assemblage p_i , une date d'arrivée des composants sur la ligne r_i et une date de livraison prévue d_i . Sachant que la ligne ne peut assembler qu'un produit à la fois, quelles sont les dates de début d'assemblage de chaque produit qui minimisent le plus grand retard de livraison ?

$$\min L$$

$$L \geq L_i \quad i = 1, \dots, n$$

$$L_i \geq C_i - d_i \quad i = 1, \dots, n$$

$$C_i \geq r_i + p_i$$

$$C_i \geq C_j + p_i - Mx_{ji} \quad i = 1, \dots, n, j = 1, \dots, n, i \neq j$$

$$x_{ij} + x_{ji} = 1$$

$$C_i \geq 0 \quad i = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, n, j = 1, \dots, n, i \neq j$$

avec M suffisamment grand (par exemple $M = \sum_{i=1}^n p_i$)

Formulation 0-1 pour le problème d'affectation

n travaux doivent être réalisés par n employés. Certaines personnes sont plus expérimentées que d'autre et le temps de réalisation par un employé i d'un travail j a une durée c_{ij} . Quelle est l'affectation qui minimise la durée moyenne des tâches ?

$$\max \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, n, j = 1, \dots, n$$

Mauvais !

Formulation 0-1 pour le problème de plus court chemin

Un réseau routier est représenté par un graphe orienté $G = (V, A)$. Le transport direct d'un lieu i vers un lieu j est possible si $(i, j) \in A$ et a une durée c_{ij} . Etant donné deux lieux s et t , quel est le chemin à coût minimum de s à t ?

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ & \sum_{k \in V^+(s)} x_{sk} - \sum_{k \in V^-(s)} x_{ks} = 1 \\ & \sum_{k \in V^+(i)} x_{ik} - \sum_{k \in V^-(i)} x_{ki} = 0 \quad i \in V \setminus \{s, t\} \\ & \sum_{k \in V^+(t)} x_{tk} - \sum_{k \in V^-(t)} x_{kt} = -1 \\ & x_{ij} \in \{0, 1\} \quad (i, j) \in A \end{aligned}$$

Mauvais !

PLNE pour le problème de flot à coût minimum

Un réseau de transport de marchandises est représenté par un graphe orienté $G = (V, A)$.

Chaque sommet $i \in V$ est associé a une valeur b_i appelée demande et correspond a un lieu de production ($b_i > 0$), de consommation ($b_i < 0$) ou de simple transit ($b_i = 0$) de marchandises.

Chaque arc $(i, j) \in A$ représente un moyen de transport de marchandises entre deux lieux limité par une capacité h_{ij} et de coût unitaire c_{ij} . Quel est le plan de transport qui satisfait toutes les demandes à coût minimal ?

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ & \sum_{k \in V^+(i)} x_{ik} - \sum_{k \in V^-(i)} x_{ki} = b_i \quad i \in V \setminus \{s, t\} \\ & x_{ij} \leq h_{ij} \quad (i, j) \in A \\ & x_{ij} \in \mathbb{Z}_+ \end{aligned}$$

Mauvais !

Décision et optimisation

Soit un problème d'optimisation discrète $\min\{f(x) : x \in \mathcal{X} \subseteq R^n\}$. Le problème de décision associé consiste à déterminer, étant donné une valeur $v \in R$, s'il existe un point $x \in \mathcal{X}$ tel que $f(x) \leq v$. On dit également que le problème d'optimisation associé est polynomial.

On détermine la complexité des problèmes d'optimisation relativement à la complexité du problème de décision associé.

Problèmes faciles (classe P de problèmes de décision)

Un problème de décision est facile s'il existe un algorithme polynomial en fonction de la taille des données pour résoudre exactement toutes les instances du problème. Si la taille des données est n , il existe un entier p tel que l'algorithme a une complexité dans le pire des cas de $O(n^p)$.

La classe NP de problèmes de décision

Soit un problème d'optimisation discrète $\min\{f(x) : x \in \mathcal{X} \subseteq R^n\}$. On dit que le problème de décision associé est dans la classe NP si, étant donné $y \in R^n$, on peut déterminer en temps polynomial si $y \in X$ et, dans le cas positif, si $f(y) \leq v$. Remarquons qu'on a $P \subseteq NP$.

Les problèmes de décision NP -complets

Il existe un ensemble de problèmes de décision appartenant à la classe NP pour lesquels aucun algorithme polynomial n'a été trouvé. Ces problèmes sont équivalents au sens où si on trouve un algorithme polynomial pour en résoudre un alors on peut en déduire des algorithmes polynomiaux pour résoudre tous les autres. On appelle ces problèmes NP -complets. Les problèmes d'optimisation associés des problèmes NP -difficiles.

Le problème de résolution d'un PLNE est NP -difficile (mais on peut formuler un problème facile comme un PLNE!)

La programmation linéaire

Un programme linéaire ne comporte aucune contrainte d'intégrité des variables $\min\{cx : A_x \leq b\}$. Résoudre un programme linéaire est un problème polynomial. On utilise en général l'algorithme du simplexe.

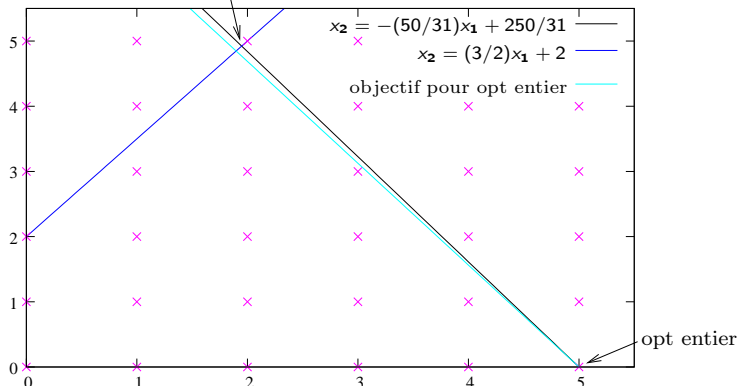
Interprétation géométrique

La PL et la PLNE ont une interprétation géométrique. En PL, l'ensemble des contraintes forme un polyèdre convexe. La fonction objectif définit un ensemble d'hyperplans. L'optimum est l'ensemble des points du polyèdre par lesquels passe l'hyperplan de valeur maximale. Si le polyèdre est borné et non vide, l'optimum de la PL est atteint sur un point extrême du polyèdre. En PLNE, l'ensemble des contraintes d'intégrité entraîne la perte de la convexité.

Comparaison des solutions entières et continues

$$\begin{aligned} \max & 1.0x_1 + 32/50x_2 \\ & 50x_1 + 31x_2 \leq 250 \\ & 3x_1 - 2x_2 \geq -4 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{aligned}$$

opt continu



L'optimum continu $(376/193, 950/193)$ de valeur 5.09 donne une solution arrondie $(2, 4)$ de valeur 4.56 très éloignée de l'optimum entier $(5, 0)$ de valeur 5.

Définition

Un problème d'optimisation $\min\{f(x) : x \in \mathcal{T} \subseteq \mathbb{R}^n\}$ est une relaxation d'un problème $\min\{g(x) : x \in \mathcal{X} \subseteq \mathbb{R}^n\}$ si $\mathcal{X} \subseteq \mathcal{T}$ et $f(x) \leq g(x)$ pour tout $x \in \mathcal{X}$.

Relaxation continue d'un PLNE

La PL obtenu par relaxation des contraintes d'intégrité d'un PLNE définit une relaxation de ce PLNE. Pour un problème de minimisation, la solution optimale du PL donne une borne inférieure de la valeur de la solution optimale du PLNE.

Preuve d'optimalité ou d'infaisabilité

Si la relaxation est infaisable, alors le problème est infaisable. Soit \tilde{x}^* la solution optimale de la relaxation. Si $f(x^*) = g(x^*)$ et si $\tilde{x}^* \in \mathcal{X}$ alors \tilde{x}^* est la solution optimale du problème. En PLNE, si la solution de la PL est entière alors la solution est optimale pour le PLNE.

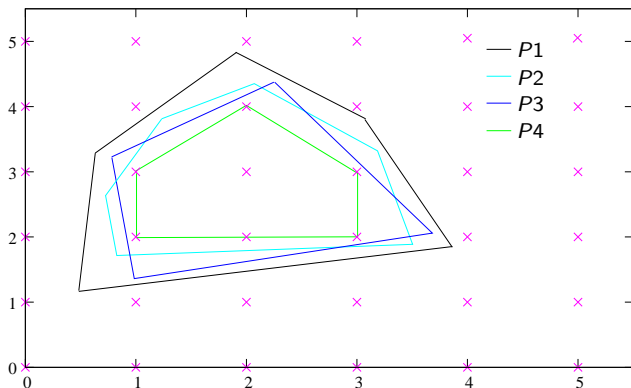
Définition

Il n'y a pas une unique formulation de PLNE pour un problème d'optimisation discrète donné. On peut comparer les différentes formulations PLNE d'un même problème relativement à la qualité de la relaxation. Soient (P1) $\min\{cx : Ax \leq b, x \in Z_+\}$ et (P2) $\min\{cx : Dx \leq e, x \in Z_+\}$ deux PLNE du même problème d'optimisation discrète. (P1) est meilleure que (P2) si $X_1 \subset X_2$ avec $X_1 = \{x : Ax \leq b\}$ et $X_2 = \{x : Dx \leq e\}$. En effet la solution de la relaxation (P1) donne une meilleure borne inférieure que la solution de la relaxation (P2).

Formulation idéale

Il existe une formulation idéale, donnée par l'enveloppe convexe du polyèdre $X = \{x : Ax \leq b, x \in Z_+\}$, notée $\text{conv}(X)$.

Bonnes et mauvaises formulations



$P4 \subset P2, P3 \subset P1 \Rightarrow P4$ est meilleure que $P1, P2$ et $P3$. $P2$ et $P3$ sont meilleures que $P1$. $P4$ est la formulation idéale $\text{conv}(\{(1, 2), (1, 3), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3)\})$.

Le problème de découpe

Un atelier de textile dispose d'un ensemble de m rouleaux de tissu, chacun étant de longueur l . L'atelier doit découper n types de pièces, chaque pièce i étant de longueur l_i et étant requise en n_i exemplaires. Quelle découpe permet d'obtenir toutes les pièces demandées en minimisant le nombre de rouleaux utilisés ?

Variables de décision pour la formulation P1

- y_k variable binaire indiquant si le rouleau k est utilisé.
- y_k variable binaire indiquant si le rouleau k est utilisé.
- x_{ik} variable entière donnant le nombre de fois que la pièce i est découpée sur le rouleau k .

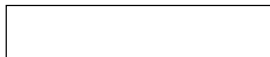
Exemple

$$m = 500$$

$$L = 100$$



$$l_1 = 45$$



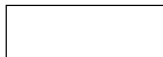
$$n_1 = 97$$

$$l_2 = 36$$



$$n_2 = 610$$

$$l_3 = 31$$



$$n_3 = 395$$

$$l_4 = 14$$



$$n_4 = 211$$

Formulation P1

$$\min \sum_{k=1}^m y_k$$

$$\sum_{k=1}^m x_{ik} \geq n_i \quad i \in \{1, \dots, n\}$$

$$\sum_{i=1}^n l_i x_{ik} \leq L y_k \quad k \in \{1, \dots, m\}$$

$$y_k \in \{0, 1\}^m \quad k \in \{1, \dots, m\}$$

$$x_{ik} \in \mathbb{Z}^+ \quad i \in \{1, \dots, n\}, k \in \{1, \dots, m\}$$

Introduction du concept de "patron"

Un patron est un mode de découpe d'un rouleau entier (exemple : 2 pièces de type 1 et 1 pièce de type 2). On peut énumérer a priori tous les patrons possibles. Les rouleaux étant identiques, il y a T patrons, tous rouleaux confondus.

1. 45	11. 36 36	21. 36 14 14 14 14	31. 14
2. 45 45	12. 36 36 14	22. 31	32. 14 14
3. 45 36	13. 36 36 14 14	23. 31 31	33. 14 14 14
4. 45 36 14	14. 36 31	24. 31 31 31	34. 14 14 14 14
5. 45 31	15. 36 31 31	25. 31 31 14	35. 14 14 14 14 14
6. 45 31 14	16. 36 31 14	26. 31 31 14 14	36. 14 14 14 14 14 14
7. 45 14	17. 36 31 14 14	27. 31 14	37. 14 14 14 14 14 14 14
8. 45 14 14	18. 36 14	28. 31 14 14	
9. 45 14 14 14	19. 36 14 14	29. 31 14 14 14	
10. 36	20. 36 14 14 14	30. 31 14 14 14 14	$T = 37$

Variables de décision pour la formulation P2

- λ_t variable entière indiquant le nombre de fois où le patron de découpe t est utilisé

Soit a_{it} le nombre de pièces de type i découpées dans le patron t .

Formulation P2

$$\begin{aligned} \min \quad & \sum_{t=1}^T \lambda_t \\ \sum_{t=1}^T \lambda_t a_{it} & \geq n_i \quad i \in \{1, \dots, n\} \\ \lambda_t & \in \mathbb{Z}^+ \quad t \in \{1, \dots, T\} \end{aligned}$$

Relaxation continue de P1

Model size: 504 constraints, 2500 variables, 4500 non-zeros.
Optimal solution 415.24 after 1470 iter. 0.383 seconds

Relaxation continue de P2

Model size: 4 constraints, 37 variables, 64 non-zeros.
Optimal solution 452.25 after 4 iter. 0.000 seconds

(Résolution avec l'algorithme du simplexe intégré dans le solveur libre `lp_solve`
(`lpsolve.sourceforge.net`)

Matrice Unimodulaire

Pour les problèmes faciles (polynomiaux), on arrive à décrire de manière compacte l'enveloppe convexe du polyèdre des contraintes. C'est notamment le cas si la matrice des contraintes est unimodulaire. Dans ce cas le problème est résolu par PL, la solution optimale étant automatiquement entière.

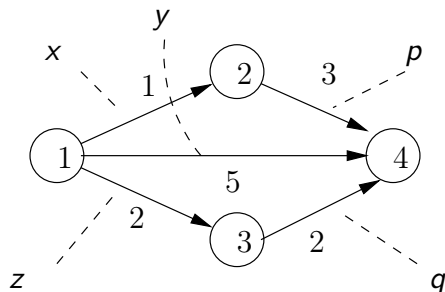
Condition suffisante

La matrice A est unimodulaire si elle ne contient que des 1, -1 ou 0, si chaque colonne contient au plus deux coefficients non nuls et s'il existe une partition M_1, M_2 des lignes telle que chaque colonne j contenant deux coefficients non nuls vérifie $\sum_{i \in M_1} a_{ij} = \sum_{i \in M_2} a_{ij}$.

Exemple de problèmes facile : plus court chemin

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ & \sum_{k \in V^+(s)} x_{sk} - \sum_{k \in V^-(s)} x_{ks} = 1 \\ & \sum_{k \in V^+(i)} x_{ik} - \sum_{k \in V^-(i)} x_{ki} = 0 \quad i \in V \setminus \{s, t\} \\ & \sum_{k \in V^+(t)} x_{tk} - \sum_{k \in V^-(t)} x_{kt} = -1 \\ & x_{ij} \in \{0, 1\} \quad (i, j) \in A \end{aligned}$$

Exemple de problème facile : plus court chemin



$$\begin{array}{rccccrcr} x & + & y & + & z & & = & 1 \\ x & & & & & -p & = & 0 \\ & & & & z & & -q & = & 0 \\ & & y & & & +p & +q & = & 0 \end{array}$$

La matrice est bien totalement unimodulaire avec $M1 = \{\text{ligne 1}\}$ et $M2 = \{\text{lignes 2,3,4}\}$.

Exemple de problème facile : plus court chemin

$$\min 4x + 5y + 4z$$

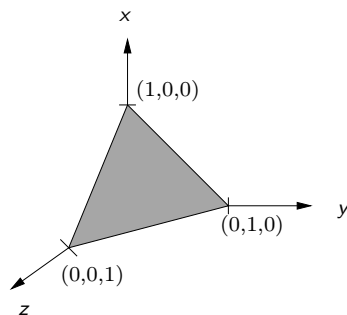
$$x + y + z = 1$$

$$0 \leq x \leq 1$$

$$0 \leq y \leq 1$$

$$0 \leq z \leq 1$$

Les points extrêmes du polyèdre des contraintes (portion du plan délimité par les points $(1,0,0)$, $(0,1,0)$ et $(0,0,1)$) sont entiers.



La contraintes d'intégrité est inutile pour les PLNE suivants

- Problème d'affectation
- Plus court chemin
- Flot à coût minimum

Ces problèmes peuvent être résolus avec l'algorithme du simplexe ou avec des algorithmes spécifiques plus efficaces.

Pour les autres problèmes, la relaxation ne donne qu'une évaluation par défaut. Deux grandes approches de résolution complémentaires existent pour obtenir la solution optimale entière.

- Génération d'inégalités valides
- Séparation et évaluation.

Problème de séparation

Etant donné un problème d'optimisation discrète $\min\{f(x) : x \in X \subseteq R^n\}$ et un point $x^* \in R^n$, déterminer si $x^* \in \text{conv}(X)$ et, sinon, générer une inégalité $\pi x \leq \pi_0$ satisfaite par tous les points de X mais violée par x^* .

En PLNE, la méthode de Gomory permet de générer toutes les inégalités valides permettant in fine d'obtenir $\text{conv}(X)$. Il s'agit de résoudre la relaxation continue du PLNE, de trouver une base optimale et de choisir une variable de base non entière et de générer une inégalité valide à partir de la contrainte associée à cette variable.

La méthode de Gomory sur un exemple

Résolvons la relaxation continue du PLNE suivant par la méthode du simplexe

$$\begin{aligned} \max x_1 + 32/50x_2 \\ 50x_1 + 31x_2 &\leq 250 \\ 3x_1 - 2x_2 &\geq -4 \\ x_1, x_2 &\in \mathbb{Z}^+ \end{aligned}$$

Ecriture sous forme canonique :

$$\begin{aligned} \max x_1 + 32/50x_2 \\ s_1 = 250 - 50x_1 - 31x_2 \\ s_2 = 4 + 3x_1 - 2x_2 \\ x_1, x_2, s_1, s_2 &\geq 0 \end{aligned}$$

La solution de base est $(0, 0, 250, 4)$. x_1 entre en base. $s_1 \geq 0 \implies x_1 \leq 5$. Le PL se réécrit :

$$\begin{aligned} \max 5 - s_1/50 + x_2/50 \\ x_1 = 5 - s_1/50 - 31/50x_2 \\ s_2 = 19 - 3/50s_1 - 193/50x_2 \\ x_1, x_2, s_1, s_2 &\geq 0 \end{aligned}$$

La solution de base est $(5, 0, 0, 19)$. x_2 entre en base. $x_1 \geq 0 \implies x_2 \leq 250/31$.
 $s_2 \geq 0 \implies x_2 \leq 950/193$

La méthode de Gomory sur un exemple (suite)

Réolvons la relaxation continue du PLNE suivant par la méthode du simplexe

$$\max 984/183 - 98/4825s_1 - s_2/193$$

$$x_1 = 376/193 - 2/193s_1 + 31/193s_2$$

$$x_2 = 950/193 - 3/193s_1 - 50/193s_2$$

$$x_1, x_2, s_1, s_2 \geq 0$$

La solution optimale du PL est $(376/193, 950/193, 0, 0)$, de valeur $984/183$.

Considérons la première contrainte. Séparons chaque coefficient en une partie entière et une partie fractionnaire positive. . On obtient :

$$(0/1 + 0)x_1 + (2/193 + 0)s_1 + (162/193 - 1)s_2 = 1 + 183/193$$

et la coupe de Gomory

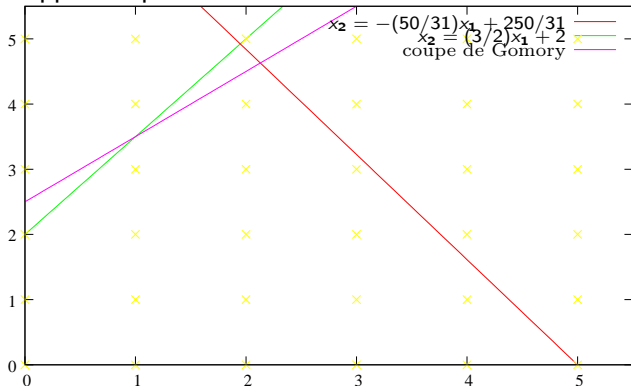
$$2/193s_1 + 162/193s_2 \geq 183/193$$

soit, en remplaçant avec les variables originales :

$$-2x_1 + 2x_2 \leq 5$$

La méthode de Gomory sur un exemple (fin)

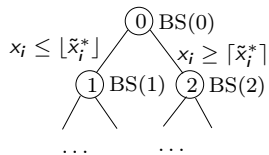
On ajoute l'inégalité ainsi générée et on résout à nouveau. La méthode stoppe lorsque la solution obtenue est entière.



Un nombre exponentiel d'inégalités valides peut être nécessaire. On couple en pratique avec une procédure de séparation et d'évaluation.

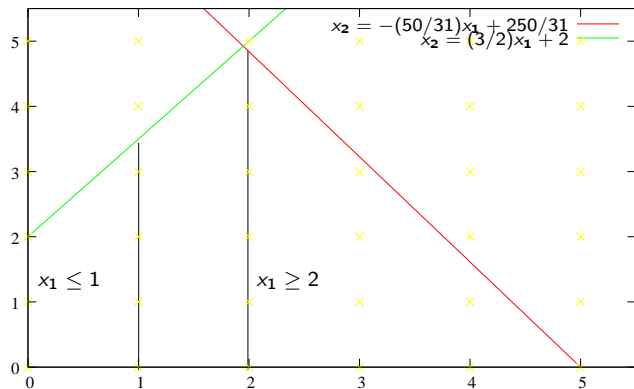
La méthode de Séparation et Evaluation (Branch and Bound)

Lorsque la relaxation continue d'un PLNE n'est pas entière (y-compris après l'ajout d'un nombre raisonnable d'inégalités valides), on peut résoudre le problème au moyen d'un arbre de recherche. En sélectionnant une variable non entière à l'optimum, par exemple x_i de valeur \tilde{x}_i^* . On sépare en deux sous-problème en considérant successivement les contraintes $x_i \leq \lfloor \tilde{x}_i^* \rfloor$ et $x_i \geq \lceil \tilde{x}_i^* \rceil$.



On applique récursivement ce principe jusqu'à trouver une solution entière. Pour un problème de maximisation, on obtient une borne inférieure BI . A chaque nœud ν de l'arbre de recherche on résout la relation continue, ce qui donne une borne supérieure $BS(\nu)$. Il est inutile de développer un nœud lorsque $BS(\nu) \leq BI$ ou lorsque la relaxation est infaisable.

Exemple de Branch and Bound



Définition

Deux problèmes d'optimisation (P) $z = \max\{c(x) : x \in X\}$ et (D) $w = \min\{w(u) : u \in U\}$ forment une paire (faiblement) duale si $c(x) \leq w(u)$ pour tout $x \in X$ et $u \in U$. Si, de plus, $z = w$ alors ils forment une paire fortement duale.

Programmes linéaires duaux

Les programmes linéaires $\max\{cx : Ax \leq b, x \in R^n+\}$ et $\min\{ub : uA \geq c, u \in R^m+\}$ forment une paire fortement duale.

Dual faible d'un PLNE

Le PLNE $\max\{cx : Ax \leq b, x \in R^n+\}$ et le PL $\min\{ub : uA \geq c, u \in R^m+\}$ forment une paire faiblement duale.

Intérêt : toute solution réalisable de (D) donne une borne supérieure de (P). Si (D) est non borné, (P) est infaisable. Si $x^* \in X$ et $u^* \in U$ vérifient $c(x^*) = w(u^*)$ alors x^* est optimal pour le primal et u^* est optimal pour le dual.

Relaxation lagrangienne

Comment se débarrasser de contraintes “ennuyeuses” ou liant plusieurs sous-problèmes indépendants ?

Relaxation lagrangienne

Soit le PLNE (P) $z = \max\{cx : Ax \leq b, Dx \leq d, x \in Z_+^n\}$. Supposons que $Dx \leq d$ constituent m contraintes. Pour tout $u = (u_1, \dots, u_m) \geq 0$, Le PLNE $z(u) = \max\{cx + u(d - Dx) : Ax \leq b, x \in Z_+^n\}$ est une relaxation de (P).

Dual lagrangien

Le dual lagrangien est le problème de recherche de la meilleure borne supérieure $w_{DL} = \min\{z(u) : u \geq 0\}$. Si la contrainte dualisée est une égalité, on minimise sans restriction de signe $w_{DL} = \min\{z(u)\}$.

Preuve d'optimalité

Si $u \geq 0$, $Dx(u) \leq d$ et si $(Dx(u))_i = d_i$ pour tout $u_i > 0$ alors $x(u)$ est optimal pour (P).

Relaxation lagrangienne, méthode de décomposition

La relaxation lagrangienne permet de résoudre indépendamment plusieurs sous-problèmes. Soit le problème de localisation :

$$\max \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} - \sum_{j=1}^n f_j y_j$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, m$$

$$x_{ij} \leq y_j \quad i = 1, \dots, m, j = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m, j = 1, \dots, n, \quad y_j \in \{0, 1\} \quad j = 1, \dots, n$$

Dualisons les contraintes $x_{ij} \leq y_j \quad i = 1, \dots, m, j = 1, \dots, n$.

$$\max \sum_{i=1}^m \sum_{j=1}^n (c_{ij} x_{ij} - u_j) - \sum_{j=1}^n f_j y_j + \sum_{i=1}^m u_i$$

$$x_{ij} \leq y_j \quad i = 1, \dots, m, j = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m, j = 1, \dots, n, \quad y_j \in \{0, 1\} \quad j = 1, \dots, n$$

Relaxation lagrangienne du problème de localisation

Il s'ensuit une décomposition en plusieurs sous-problèmes :

$$z(u) = \sum_{j=1}^n z_j(u) + \sum_{i=1}^n u_i \text{ avec}$$

$$z_j(u) = \max \sum_{i=1}^m \sum_{j=1}^n (c_{ij}x_{ij} - u_i) - \sum_{j=1}^n f_j y_j$$

$$x_{ij} \leq y_i \quad i = 1, \dots, m, j = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m, j = 1, \dots, n$$

$$y_j \in \{0, 1\} \quad j = 1, \dots, n$$

$$f = (2, 4, 5, 3, 3) \quad c = \begin{pmatrix} 6 & 2 & 1 & 3 & 5 \\ 4 & 10 & 2 & 6 & 1 \\ 3 & 2 & 4 & 1 & 3 \\ 2 & 0 & 4 & 1 & 4 \\ 1 & 8 & 6 & 2 & 5 \\ 3 & 2 & 4 & 8 & 1 \end{pmatrix} \quad u = (5, 6, 3, 2, 5, 4)$$

On obtient la matrice des coûts révisés suivante :

$$f = (2, 4, 5, 3, 3) \quad c - u = \begin{pmatrix} 1 & -3 & -4 & -2 & 0 \\ -2 & 4 & -4 & 0 & -5 \\ 0 & -1 & 1 & -2 & 0 \\ 0 & -2 & 2 & -1 & 2 \\ -4 & 3 & 1 & -3 & 0 \\ -1 & -2 & 0 & 4 & -3 \end{pmatrix}$$

On trouve facilement les solutions des n problèmes $z_j(u)$. $y_1 = y_3 = y_5 = 0$, $y_2 = x_{22} = x_{52} = 1$ et $y_4 = x_{64} = 1$ avec $z(u) = 3 + 1 + \sum_{i=1}^m u_i = 29$ (borne supérieure).

Il est également possible d'obtenir une borne inférieure en adaptant la solution de la relaxation lagrangienne pour qu'elle devienne réalisable. Ici les clients 1, 3 et 4 ne sont pas couverts. En prenant en plus $x_{14} = x_{32} = x_{44} = 1$ on obtient une solution réalisable de coût 25.

Relaxation lagrangienne du problème du voyageur de commerce symétrique (1/3)

Observation : Un 1-arbre est un arbre reliant les sommets $\{2, \dots, n\}$ plus deux arcs adjacents au nœud 1. Une solution réalisable du PVC est un 1-arbre dans lequel tous les sommets ont un degré égal à 2.

$$\begin{aligned} z &= \min \sum_{e \in E} c_e x_e \\ \sum_{e \in \delta(i)} x_e &= 2 \quad i \in V \\ \sum_{e \in E(S)} x_e &\leq |S| - 1 \quad 2 \leq |S| \leq |V| - 1 \\ x &\in \{0, 1\}^{|E|} \end{aligned}$$

La recherche d'un 1-arbre de poids minimal est un problème facile.
Comment faire apparaître de tels sous-problèmes ?

Relaxation lagrangienne du problème du voyageur de commerce symétrique (2/3)

En sommant $\sum_{e \in \delta(i)} x_e = 2$ pour $i \in V$ on obtient $\sum_{e \in E} x_e = n$. On remarque ainsi que la moitié des contraintes d'élimination des sous-tours sont redondantes. En effet, il n'y a pas de sous-tour dans S si et seulement s'il n'y en a pas dans $V \setminus S$. On peut donc abandonner toutes les contraintes de sous-tours telles que $1 \in S$. On dualise les contraintes de degré tout en conservant celle concernant le nœud 1. On obtient $(P(u))$:

$$z(u) = \min \sum_{e=(i,j) \in E} (c_e - u_i - u_j) x_e + 2 \sum_{i \in V} u_i$$

$$\sum_{e \in \delta(1)} x_e = 2$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \quad 2 \leq |S| \leq |V| - 1, 1 \notin S$$

$$\sum_{e \in E} x_e = n$$

$$x \in \{0, 1\}^{|E|}$$

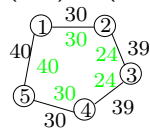
Les solutions réalisables de $P(u)$ sont précisément des 1-arbres.

Relaxation lagrangienne du problème du voyageur de commerce symétrique (3/3)

Considérons le PVCS ci-dessous et des multiplicateurs de Lagrange $u = (0, 0, -15, 0, 0)$.

$$(c_e) = \begin{pmatrix} - & 30 & 26 & 50 & 40 \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix} \quad (c_e - u_i - u_j) = \begin{pmatrix} - & 30 & 41 & 50 & 40 \\ - & - & 39 & 40 & 50 \\ - & - & - & 39 & 41 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix}$$

$z(u)$ est obtenue en prenant les arcs issus de 1 de valeur minimale plus la solution optimale de l'arbre de poids minimal dans $\{2, 3, 4, 5\}$ qui donne les arcs $(4, 5)$, $(2, 3)$ et $(3, 4)$. On obtient $z(u) = 178 - 30 = 148$



La solution relâchée est également un tour de longueur 148 (solution optimale de (P)).

Force du dual lagrangien (1/2)

$z(u) = \max\{cx + u(d - Dx) : Ax \leq b, x \in Z_+^n\}$. Posons
 $X = \{Ax \leq b, x \in Z_+^n\} = \{x^1, \dots, x^T\}$.

$$\begin{aligned}w_{DL} &= \min_{u \geq 0} z(u) \\ &= \min_{u \geq 0} \{ \max_{x \in X} [cx + u(d - Dx)] \} \\ &= \min_{u \geq 0} \{ \max_{t=1, \dots, T} [cx^t + u(d - Dx^t)] \} \\ &= \min \{ \nu : \nu \geq cx^t + u(d - Dx^t), t = 1, \dots, T, u \in R_+^m, \nu \in R \}\end{aligned}$$

qui est un programme linéaire dont le dual est

$$w_{DL} = \max \sum_{t=1}^T \mu_t (cx^t)$$

$$\sum_{t=1}^T \mu_t (Dx^t - d) \leq 0$$

$$\sum_{t=1}^T \mu_t = 1$$

$$\mu_t \in R_+^T$$

Or $\text{conv}(X) = \{x = \sum_{t=1}^T \mu_t x^t, \sum_{t=1}^T \mu_t, \mu \in R_+^T\}$. Ainsi on obtient :

$$w_{DL} = \max(cx)$$

$$Dx \leq d$$

$$x \in \text{conv}(X)$$

Dans le cas où $X = \{x \in Z_+^n : Ax \leq b\}$ et $\text{conv}(X) = \{x \in R_+^n : Ax \leq b\}$, alors w_{DL} est égal à la relaxation continue de (P) . Sinon, on a $w_{DL} \leq \max\{cx : Ax \leq b, Dx \leq d, x \in R_+^n\}$.

Résolution du dual lagrangien

Le problème du dual lagrangien peut être vu comme un problème de minimisation d'une fonction linéaire par morceau convexe mais non différentiable $z(u)$. On utilise l'algorithme du sous gradient.

Algorithme du sous-gradient

- 1 Initialiser $u \leftarrow u^0$
- 2 Résoudre la la relaxation lagrangienne $P(u^k)$ de solution optimale $x(u^k)$
- 3 $u^{k+1} \leftarrow \max\{u^k - \mu_k(d - Dx(u^k)), 0\}$
- 4 $k \leftarrow k + 1$; retourner à l'étape 2.

où μ_k est la longueur du pas.

La règle principale pour que l'algorithme converge vers la solution optimale est que la série μ_k doit diverger $\sum_k \mu_k \rightarrow \infty$ alors que $\mu_k \rightarrow 0$ quand $k \rightarrow \infty$. Pour accélérer la convergence, on peut prendre $\mu_k = \epsilon_k [z(u^k) - \bar{w}] / \|d - Dx(u^k)\|^2$ avec $0 < \epsilon_k < 2$ et $\bar{w} \geq w_{DL}$. Si on ne connaît de borne supérieure, on peut prendre une borne inférieure $\underline{w} \leq w_{DL}$ obtenue par une heuristique.

Résolution du dual lagrangien pour le PVCS (1/3)

La direction du pas est donnée par

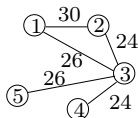
$$u_i^{k+1} = u_i^k + \mu_k(2 - \sum_{e \in \delta(i)} x_e(u^k))$$

La taille du pas est donnée par

$$\mu_k = \epsilon_k(\underline{w} - z(u^k)) / \sum_{i \in V} (2 - \sum_{e \in \delta(i)} x_e(u^k))^2$$

Supposons qu'on ait obtenu une borne supérieure de 148 par une heuristique mais qu'on ne connaisse pas de borne inférieure. On a donc $\bar{w} = 148$. On prend $\epsilon_k = 1$

Itération 1 : $u^1 = (0, 0, 0, 0, 0)$. la matrice des coûts révisée est c . L'arbre optimal est



avec $z(u^1) = 130$. On a $(2 - \sum_{e \in \delta(i)} x_e(u^k)) = (0, 0, -2, 1, 1)$. On obtient donc

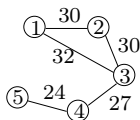
$$u^2 = u^1 + [(148 - 130)/6](0, 0, -2, 1, 1)$$

Itération 2 : $u^2 = (0, 0, -6, 3, 3)$. La matrice des coûts révisée est

$$(c_e - u_i^2 - u_j^2) = \begin{pmatrix} - & 30 & 32 & 47 & 37 \\ - & - & 30 & 37 & 47 \\ - & - & - & 27 & 29 \\ - & - & - & - & 24 \\ - & - & - & - & - \end{pmatrix}$$

On obtient $z(u^2) = 143 + 2 \sum_i u_i^2 = 143$ et

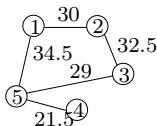
$u^3 = u^2 + ((148 - 143)/2)(0, 0, -1, 0, 1)$. Le nouvel arbre optimal est :



Itération 3 : $u^3 = (0, 0, -17/2, 3, 11/2)$. La matrice des coûts révisée est

$$(c_e - u_i^2 - u_j^2) = \begin{pmatrix} - & 30 & 34.5 & 47 & 34.5 \\ - & - & 32.5 & 37 & 44.5 \\ - & - & - & 29.5 & 29 \\ - & - & - & - & 21.5 \\ - & - & - & - & - \end{pmatrix}$$

Le nouvel arbre optimal est :



On obtient $z(u^3) = 147.5$ et comme la solution doit être entière, il s'ensuit que $z \geq \lceil 147.5 \rceil = 148$. On prouve donc l'optimalité de la borne supérieure de départ.

Génération de colonnes

Certaines formulations de PLNE comprennent un nombre très important de variables. Leur intérêt est souvent de correspondre à une meilleure relaxation qu'une formulation compacte.

PLNE alternatif pour le problème de localisation

$$\min \sum_{j=1}^n \sum_{S \neq \emptyset} (\sum_{i \in S} c_{ij} + f_j) \lambda_S^j$$

$$\sum_{j=1}^n \sum_{S \neq \emptyset, i \in S} \lambda_S^j = 1 \quad i = 1, \dots, m$$

$$\sum_{S \neq \emptyset} \lambda_S^j \leq 1 \quad j = 1, \dots, n$$

$$\lambda_S^j \in \{0, 1\} \quad \emptyset \neq S \subseteq \{1, \dots, m\}, j = 1, \dots, n$$

Les algorithmes de génération de colonnes permettent de résoudre de manière efficace de tels PLNE.

Reformulation de Dantzig-Wolfe d'un PLNE (1/2)

De telles formulations peuvent être obtenues par la reformulation de Dantzig-Wolfe. Soit

$$z = \max \left\{ \sum_{k=1}^K c^k x^k : \sum_{k=1}^K A^k x^k = b, x^k \in X^k \quad k = 1, \dots, K \right\}$$

un PLNE dont l'espace des solutions réalisable est décomposable en $K \geq 1$ sous-espaces $X^k = \{x \in Z_+^{n_k} : D^k x^k \leq d_k\}$ dont les variables sont liées uniquement par les contraintes de liaison $A^k x^k = b$.

On note $X^k = \{x^{k,t}\}_{t=1}^{T_k}$ comme un ensemble fini mais très grand de points. Alors

$$X^k = \{x^k \in Rn_k : x^k = \sum_{t=1}^{T_k} \lambda_{k,t} x^{k,t}, \sum_{t=1}^{T_k} \lambda_{k,t} = 1, \lambda_{k,t} \in 0, 1 \quad t = 1, \dots, T_k\}$$

Reformulation de Dantzig-Wolfe d'un PLNE (2/2)

On substitue x^k par $\sum_{t=1}^{T_k} \lambda_{k,t} x^{k,t}$ dans

$$z = \max \left\{ \sum_{k=1}^K c^k x^k : \sum_{k=1}^K A^k x^k = b, x^k \in X^k \quad k = 1, \dots, K \right\}$$

et on obtient le PLNE appelé problème maître entier :

$$z = \max \left\{ \sum_{k=1}^K \sum_{t=1}^{T_k} (c^k x^{k,t}) \lambda_{k,t} \right.$$

$$\left. \sum_{k=1}^K \sum_{t=1}^{T_k} (A^k x^{k,t}) \lambda_{k,t} = b \right.$$

$$\left. \sum_{t=1}^{T_k} \lambda_{k,t} = 1 \quad k = 1, \dots, K \right.$$

$$\left. \lambda_{k,t} \in 0, 1 \quad t = 1, \dots, T_k, k = 1, \dots, K \right\}$$

Formulation de PLNE Compacte

$$\begin{aligned} \min \quad & \sum_{k=1}^m y_k \\ \text{s.t.} \quad & \sum_{k=1}^m x_{ik} \geq n_i \quad i \in \{1, \dots, n\} \\ & \sum_{i=1}^n l_i x_{ik} \leq L y_k \quad k \in \{1, \dots, m\} \\ & y_k \in \{0, 1\}^m \quad k \in \{1, \dots, m\} \\ & x_{ik} \in \mathbb{Z}^+ \quad i \in \{1, \dots, n\}, k \in \{1, \dots, m\} \end{aligned}$$

Soit

$$X^k = \{(x_{1k}, \dots, x_{nk}, y_k) \in \mathbb{Z}^+ \times \dots \times \mathbb{Z}^+ \times \{0, 1\} : \sum_{i=1}^n l_i x_{ik} \leq L y_k\}.$$

Formulation équivalente

$$\min \sum_{k=1}^m y_k$$

$$\sum_{k=1}^m x_{ik} \geq n_i \quad i \in \{1, \dots, n\}$$

$$(x_{1k}, \dots, x_{nk}, y_k) \in X^k$$

$$y_k \in \{0, 1\}^m \quad k \in \{1, \dots, m\}$$

$$x_{ik} \in \mathbb{Z}^+ \quad i \in \{1, \dots, n\}, k \in \{1, \dots, m\}$$

avec

$$X^k = \{(x_{1k}, \dots, x_{nk}, y_k) \in \mathbb{Z}^+ \times \dots \times \mathbb{Z}^+ \times \{0, 1\} : \sum_{i=1}^n l_i x_{ik} \leq L y_k\}.$$

Reformulation de Dantzig-Wolfe du problème de découpe

X_k contient un nombre potentiellement très grand mais fini de points. Soit $T_k + 1$ ce nombre de points. On a

$$X_k = \{(a_{1kt}, \dots, a_{nkt}, b_{kt})\}_{t=0}^{T_k}$$

On suppose que $t = 0$ correspond au point tel que $y_k = 0$ et donc pour tout i , $a_{1k0} = a_{2k0} = \dots = a_{nk0} = b_{k0} = 0$ alors que, $1 \geq t \geq T_k$ correspond aux points tels que $y_k = 1$. Les valeurs des variables x_{ik} et y_k sont déterminées par la sélection d'un point dans X_k . Le problème de sélection d'un point dans X_k est représenté par un ensemble de variables de décision $\lambda_{kt} \in \{0, 1\}$, $t = 0, \dots, T_k$ telles que $\lambda_{kt} = 1$ si on sélectionne le point $(a_{1kt}, \dots, a_{nkt}, b_{kt})$ et $\lambda_{kt} = 0$ sinon. En imposant $\sum_{t=0}^{T_k} \lambda_{kt} = 1$, on a donc

$$x_{ik} = \sum_{t=0}^{T_k} \lambda_{kt} a_{ikt} \text{ et } y_k = \sum_{t=1}^{T_k} \lambda_{kt} b_{kt}$$

Puisque $a_{ik0} = 0$ pour tout i , $b_{k0} = 0$ et $b_{kt} = 1$ pour tout $t \geq 1$, on obtient

$$x_{ik} = \sum_{t=0}^{T_k} \lambda_{kt} a_{ikt} \text{ et } y_k = \sum_{t=1}^{T_k} \lambda_{kt}$$

Problème de découpe

On substitue $x_{ik} = \sum_{t=1}^{T_k} \lambda_{kt} a_{ikt}$ et $y_k = \sum_{t=1}^{T_k} \lambda_{kt}$.

Problème maître entier

$$\min \sum_{k=1}^m \sum_{t=1}^{T_k} \lambda_{kt}$$

$$\sum_{k=1}^m \sum_{t=1}^{T_k} \lambda_{kt} a_{ikt} \geq n_i \quad i \in \{1, \dots, n\}$$

$$\sum_{t=0}^{T_k} \lambda_{kt} = 1 \quad k \in \{1, \dots, m\}$$

$$\lambda_{kt} \in \{0, 1\} \quad k \in \{1, \dots, m\}, \quad t \in \{0, \dots, T_k\}$$

Simplification 1 : On supprime la variable λ_{k0} .

Problème maître entier

$$\min \sum_{k=1}^m \sum_{t=1}^{T_k} \lambda_{kt}$$

$$\sum_{k=1}^m \sum_{t=1}^{T_k} \lambda_{kt} a_{ikt} \geq n_i \quad i \in \{1, \dots, n\}$$

$$\sum_{t=1}^{T_k} \lambda_{kt} \leq 1 \quad k \in \{1, \dots, m\}$$

$$\lambda_{kt} \in \{0, 1\} \quad k \in \{1, \dots, m\}, \quad t \in \{1, \dots, T_k\}$$

Problème de découpe

Simplification 2 : On remarque que les ensembles X_k sont tous identiques car les rouleaux sont identiques. On a donc $T_1 = T_2 = \dots = T_k = T$ et $a_{j1t} = a_{j2t} = \dots = a_{jmt} = a_{jt}$. On introduit la variable $\lambda_t = \sum_{k=1}^m \lambda_{kt}$.

Problème maître entier

$$\min \sum_{t=1}^T \lambda_t$$

$$\sum_{t=1}^T \lambda_t a_{it} \geq n_i \quad i \in \{1, \dots, n\}$$

$$\sum_{t=1}^T \lambda_t \leq m$$

$$\lambda_t \in \mathbb{Z}^+ \quad t \in \{1, \dots, T\}$$

Simplification 3 : On supprime la contrainte $\sum_{t=1}^T \lambda_t \leq m$.

Problème maître entier (MP)

$$\min \sum_{t=1}^T \lambda_t$$

$$\sum_{t=1}^T \lambda_t a_{it} \geq n_i \quad i \in \{1, \dots, n\}$$

$$\lambda_t \in Z^+ \quad t \in \{1, \dots, T\}$$

Problème maître entier (MP)

$$z = \max \left\{ \sum_{k=1}^K \sum_{t=1}^{T_k} (c^k x^{k,t}) \lambda_{k,t} \right.$$

$$\left. \sum_{k=1}^K \sum_{t=1}^{T_k} (A^k x^{k,t}) \lambda_{k,t} = b \right.$$

$$\left. \sum_{t=1}^{T_k} \lambda_{k,t} = 1 \quad k = 1, \dots, K \right.$$

$$\left. \lambda_{k,t} \in 0, 1 \quad t = 1, \dots, T_k, k = 1, \dots, K \right.$$

La relaxation continue du problème maître entier est

$$z^{LPM} = \max \left\{ \sum_{k=1}^K c^k x^k : \sum_{k=1}^K A^k x^k = b, x^k \in \text{conv}(X^k) \quad k = 1, \dots, K \right\}$$

On a donc $z^{LPM} = w_{LD}$ avec w_{LD} solution optimale du dual lagrangien obtenu en linéarisant les contraintes liantes $\sum_{k=1}^K A^k x^k = b$.

Algorithme de génération de colonnes

- 1 Initialiser chaque ensemble \tilde{X}^k avec un nombre réduit d'éléments.
- 2 résoudre le problème maître restreint à ces ensembles de colonnes. On obtient une solution primale optimale $\tilde{\lambda}^*$ et une solution duale $(\pi, \mu) \in R^m \times R^K$.
- 3 Résoudre le problème de recherche de la colonne de cout réduit positif comme un problème d'optimisation (sous problème)

$$\zeta_k = \max\{(c^k - \pi A^k)x - \mu_k : x \in X^k\} \quad k = 1, \dots, K$$

- 4 Si $\zeta_k = 0$ pour $k = 1, \dots, K$, $\tilde{\lambda}^*$ est optimale pour *LPM*.
- 5 Sinon, si $\zeta_k > 0$ pour un k , ajouter la colonne correspondant à la solution optimale \tilde{x}^k du sous-problème à \tilde{X}^k et revenir à l'étape 2.

Borne supérieure

$$z^{LPM} \leq \pi b + \sum_{k=1}^K \mu_k + \sum_{k=1}^K \zeta_k$$

D'après la résolution du sous-problème il vient

$$\zeta_k \geq (c^k - \pi A^k)x - \mu_k \forall x \in X^k$$

D'où

$$(c^k - \pi A^k)x - \mu_k - \zeta_k \leq 0 \forall x \in X^k$$

et donc $(\pi, \mu + \zeta)$ est une solution réalisable du dual de LPM de coût $\pi b + \sum_{k=1}^K \mu_k + \sum_{k=1}^K \zeta_k \geq z^{LPM}$

Condition d'arrêt

Si les solutions des sous-problèmes $(\tilde{x}^1, \dots, \tilde{x}^K)$ satisfont les contraintes liantes alors $(\tilde{x}^1, \dots, \tilde{x}^K)$ est optimale pour LPM

On s'intéresse à la relaxation continue de (LP).

Problème maître entier (LPMP)

$$\min \sum_{t=1}^T \lambda_t$$

$$\sum_{t=1}^T \lambda_t a_{it} \geq n_i \quad i \in \{1, \dots, n\}$$

$$\lambda_t \in R^+ \quad t \in \{1, \dots, T\}$$

On considère un ensemble réduit de colonnes \mathcal{T}' .

Problème maître restreint (LPRMP)

$$\begin{aligned} \min \quad & \sum_{t \in \mathcal{T}'} \lambda_t \\ \sum_{t \in \mathcal{T}'} \lambda_t a_{it} & \geq n_i \quad i \in \{1, \dots, n\} \\ \lambda_t & \in R_+ \quad t \in \mathcal{T}' \end{aligned}$$

Soit $\tilde{\lambda}^*$ la solution optimale du problème restreint relâché et soit $\tilde{\pi}^*$ la solution du dual correspondante.

$\tilde{\lambda}^*$ est la solution optimale de (LPMP) si $\tilde{\pi}^*$ ne viole aucune contrainte du dual de (LPMP).

Dual du problème maître (DLPMP)

$$\begin{aligned} \max \quad & \sum_{i=1}^n \pi_i n_i \\ \sum_{i=1}^n \pi_i a_{it} & \leq 1 \quad t \in \{1, \dots, T\} \\ \pi_i & \in R^+ \quad i \in \{1, \dots, n\} \end{aligned}$$

Une contrainte du dual est violée si pour une colonne $t \in \{1, \dots, T\} \setminus \mathcal{T}'$, $\sum_{i=1}^n \tilde{\pi}_i^* a_{it} - 1 > 0$.

Plutôt que d'énumérer tous les points de $\{1, \dots, T\} \setminus \mathcal{T}'$, on résout le sous-problème de recherche du point $(a_i)_{i=1}^n$ de plus grand $\sum_{i=1}^n \tilde{\pi}_i^* a_{it} - 1$.

Sous-problème

$$\begin{aligned} \max \quad & \sum_{i=1}^n \tilde{\pi}_i^* a_{it} - 1 \\ \sum_{i=1}^n l_i a_i & \leq L \\ a_i & \in \mathbb{Z}_+ \end{aligned}$$

On obtient un problème de sac-à-dos entier. Si la solution optimale a^* a une valeur positive, on ajoute la colonne correspondante au problème maître restreint et on réitère le processus. Sinon, on a obtenu la solution optimale du problème maître relâché.